## REMARKS

The foregoing amendments are made to cancel claims and reduce issues for appeal rather than to avoid prior art. Applicant respectfully request reconsideration of the above identified application. Claims 1-30 and 35-46 are pending. Claims 1-30 and 35-46 are rejected. Claims 9-12, 35, and 40-46 are canceled.

Applicant respectfully notes that in the Final Office Action mailed on September 22, 2005, remarks with regard to Applicant's arguments of January 21, 2005 and of June 8, 2005; interpretations or characterizations by the Examiner, include inferences and/or potential limitations, to which Applicant does not agree.

The remaining comments are directed primarily to the rejected claims.

### 35 U.S.C. § 112 Rejections

The Final Office Action mailed on September 22, 2005, rejects Claims 1-8 and 13-30 under 35 U.S.C. 112, second paragraph, as allegedly failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention. Applicant respectfully disagrees.

The issue of definiteness is whether, in light of the teachings of the prior art and of the particular invention, the claims set out and circumscribe a particular area with a reasonable degree of precision and particularity. *In re Moore*, 439 F.2d 1232, 1235, 169 USPQ 236, 238 (CCPA 1971).

Claim 1, for example, sets forth:

> 1. (Original) An operating-system transparent method for sharing virtual address translations comprising:
> accessing a virtual address translation; and
> transparently identifying if the virtual address translation is sharable.

Applicant refers to a definition from McGraw-Hill Dictionary of Scientific and Technical Terms, Sixth Edition, (attached) which defines:

> **transparent** [COMPUT SCI] pertaining to a device or system that processes data without the user being aware of or needing to understand its operation.

The Final Office states that Applicant's response gives a dictionary definition of transparent without stating whether this definition is intended as an addition to, or instead of, specification descriptions previously cited and questioned. In response, Applicant again respectfully submits that the Office Action mailed on July 21, 2004, included inferences and/or potential limitations, to which Applicant does not agree. The Examiner points to two phrases (one from page 7 and one from page 17) and states that it is not clear whether either or both these definitions must be met.

Applicant again respectfully intends that "operating-system transparent" and "transparently identifying if the virtual address translation is sharable," be given the broadest reasonable interpretation consistent with the specification when analyzing the scope of the claim. Applicant also respectfully notes: "[t]hat claims are interpreted in light of the specification does not mean that everything in the specification must be read into the claims." *Raytheon Co. v. Roper Corp.*, 724 F.2d 951, 957, 220 USPQ 592, 597 (Fed. Cir. 1983).

Applicant has respectfully submitted the above referenced definition from McGraw-Hill Dictionary of Scientific and Technical Terms as evidence that operating-system transparent is not indefinite, being composed of well known terms of art.

Applicant does not intend that such submissions be used to narrow the scope of the claims beyond the broadest reasonable interpretation consistent with the specification. Nor should the inference be made that such submissions were made with reference to any art cited by the Examiner. Applicant is respectfully arguing that "operating-system transparent" is not indefinite.

In light of the prior art and of the particular invention, Applicant respectfully submits that "transparently identifying if the virtual address translation is sharable," sets out and circumscribes a particular area with a reasonable degree of precision and particularity in accordance with 35 U.S.C. 112, second paragraph.

Claim 13 sets forth:

13. (Original) A processing system providing operating-system transparent sharing of virtual address translations, the processing system comprising:
a first logical processor;
a second logical processor;
a storage location to store a virtual address translation; and
a control logic to access a first virtual address translation for the first logical processor in the storage location and to transparently produce a first sharing indication if the virtual address translation may be shared with the second logical processor.

Claim 20 also sets forth:

20. (Original) An apparatus to provide operating-system transparent sharing of virtual address translations, the apparatus comprising:

a control logic to access a first virtual address translation for a first processor,

the control logic further to transparently provide a first sharing indication if the first virtual address translation may be shared with a second processor.

Applicant respectfully submits that in light of the teachings of the prior art and of the particular invention, claims 13 and 20 both set forth apparatus to provide operating-system transparent (both being well known terms of art) sharing of virtual address translations comprising control logic to transparently produce or provide a sharing indication if a virtual address translation may be shared with a reasonable degree of precision and particularity in accordance with 35 U.S.C. 112, second paragraph.

The Final Office Action also rejects Claims 1-8 and 13-30 under 35 U.S.C. § 112, first paragraph, as allegedly failing to comply with the enablement requirement. Applicant respectfully disagrees and represents that the specification is in compliance with the first paragraph of § 112.

Applicant again respectfully submits that the Office Action mailed on July 21, 2004 and subsequently, the Final Office Action fail to meet the initial burden to establish a reasonable basis to question enablement. *In re Wright*, 999 F.2d 1557, 1562, 27 USPQ2d 1510, 1513 (Fed. Cir. 1993)

According to MPEP § 2164.04, before any analysis of enablement can occur, it is necessary for the Examiner to construe the claim. As stated above with regard to the second paragraph of § 112, the manner in which the Examiner extracts phrases from the specification while ignoring the context in which they appear is inconsistent with the specification and therefore incorrect when analyzing the scope of the claim. Thus, by construing the claim in a manner inconsistent with the specification, the Examiner concludes that the specification does not enable the claim.

Applicant respectively submits that the Examiner can make no prima facie case for lack of enablement by improperly construing the claims to be inconsistent with the specification.

The Office Action mailed on July 21, 2004 says that the present application does not show how to make the identification of sharability of a translation transparent to the operating

system, stating that since the PID of Fig. 8 is disclosed as a logical processor, it would not necessarily be generated by hardware and may be set by the process or some other software mechanism. The Examiner then concludes that the disclosure has not shown how to prevent a requirement to modify the operating system to set and submit the PID.

The claimed invention does not require operating systems to be prevented from setting and submitting a PID as inferred by the Examiner, but rather sets forth at least one operating-system transparent method for transparently identifying if a virtual address translation is sharable.

The Examiner has also pointed out that how the PID is generated is apparently not disclosed. Applicant has responded that the specification need not disclose what is well-known in the art and preferably omits that which is well-known to those skilled and already available to the public. *In re Buchkner*, 929 F.2d 660, 661, 18 USPQ2d 1331, 1332, (Fed. Cir. 1991).

For example, the specification discloses that (p. 14-16, par. 37-40, emphasis supplied):

> A variation of multiprocessing is known in the art as multithreading. In multithreading, multiple logical processors, which may comprise a single physical processor or multiple physical processors, perform tasks concurrently.

Applicant has referred the Examiner to an article submitted in an IDS, "On the Performance of A Multi-Threaded RISC Architecture," by Scott Lindsay and Bruno Preiss, of September 1993, which indicates that thread identifiers in multithreading processors were well-known in the art prior to the time of filing the present application (p. 5, lines 42-43).

The Office Action mailed on July 21, 2004 says that sharing indications must be compared with some input in order to determine if the input comes from a processor that is allowed to share the translation and that the only mechanism shown for this is in Fig. 8 at PID (p. 21, par. 0052). Applicant respectfully submits that the above characterization is oversimplified, at most partially correct, and only in a trivial aspect of the disclosure. For example, the logical processors $P_0$ and $P_1$ from the example shown in Figure 8, may be represented by 0 and 1 respectively without regard to any operating system issues.

The Final Office Action argues that either a PID is set by the operating system or is transparent to the operating system in both the present application and in the cited references (p. 3, lines 5-7). Applicant respectfully disagrees with this assertion, but more importantly, the Examiner's argument has little if any bearing on the issue of enablement. The Final Office Action further states that if the PID is set by an operating system, then the claims are not enabled

(p. 9, par. 25). Applicant respectfully submits that the claim sets forth transparently identifying if the virtual address translation is sharable, not setting a PID.

Applicant respectfully submits that, with regard to transparently identifying if the virtual address translation is sharable, the specification has set forth a full and clear description of the claimed subject matter to enable one skilled in the art to make and use the claimed invention. For example, the specification discloses (p. 14-16, par. 37-40, emphasis supplied):

> Control logic 604 may use the data portion 614, sharing indication 619, and data portion 624 to identify if the virtual address translation is sharable. For example, if a processor initiates a TLB request to look up a virtual address translation and the TLB entry in latches 633 and 637 contains an ASID that matches the ASID for the virtual address to be translated, and further if the entry contains a VAD that matches the VAD for the virtual address, and finally if sharing indication 619 indicates a set of logical processes including one associated with the processor initiating the TLB request, then the entry in latch 633 and latch 637 may be used to translate the virtual address. Otherwise, control logic 604 may initiate installation of a new virtual address translation entry for TLB 602.
>
> Whenever a miss occurs in TLB 602, the physical address data and other TLB data may be recovered from page tables in main memory. For one alternative embodiment control logic 604 may comprise a mechanism for recovering such data. Most modern processors use a mechanism called a page walker to access page tables in memory and compute physical addresses on TLB misses.
>
> If a processor, either directly through software or indirectly through control logic 604, initiates a TLB request to installation of a new virtual address translation entry, the TLB 602 may be searched for any existing entries that can be shared. An entry retrieved from tag array 631 and translation array 635 may then be latched by latch 633 and latch 637 respectively. If the TLB entry in latches 633 and 637 contains an ASID that matches the ASID for the virtual address to be translated, and further if the entry contains a VAD that matches the VAD for the virtual address, and finally if sharing indication 619 indicates a shared status, then the entry in latch 633 and latch 637 may be installed for the processor initiating the TLB request by adding the logical process associated with the initiating processor to the set of logical processes indicated by sharing indication 619 and thereafter the TLB entry may be used to translate the virtual address. Otherwise, control logic 604 may initiate allocation of a new virtual address translation entry for TLB 602.
>
> If a processor, either directly through software or indirectly through control logic 604, initiates a TLB request to allocate a new virtual address translation entry, the TLB 602 may be searched for any invalid or replaceable entries. The retrieved TLB entry may then be reset by control logic 604 to contain an ASID that matches the ASID for the virtual address to be translated, a VAD that matches the VAD for the virtual address, a PAD that matches the PAD of the translated physical address, an ATRD that matches the ATRD of the translated physical address, and any other associated data corresponding to the virtual address translation. Finally the entry may be installed for the processor initiating the TLB allocation request by initializing the set of logical processes indicated by sharing indication 619 to contain only the logical process associated with the initiating processor. It will be appreciated that the sharing indication 619 may be conveniently initialized by default to indicate a shared status for the virtual address translation. Alternatively if the allocation was initiated through software, for example, control logic 604 may initialize the sharing indication 619 by default to indicate a private status for the virtual address translation.

Therefore, Applicant respectfully intends that one of skill in the art would be able to transparently identify if the virtual address translation is sharable without undo experimentation. The present specification further states with regard to Fig. 7a (p. 19, par. 46-48, emphasis supplied):

> In Figure 7a, for example, a sharing indication corresponding to virtual address translation entry 711 indicates a private status of P and a set of logical processes of 0001, the low order bit being set to indicate that entry 711 may be used exclusively to translate virtual addresses for processor 710. Similarly a sharing

indication corresponding to virtual address translation entry 713 indicates a private status of P and a set of logical processes of 0100, indicating that entry 713 may be used exclusively to translate virtual addresses for processor 740.

A sharing indication corresponding to virtual address translation entry 712 indicates a shared status of S and a set of logical processes of 0101, indicating that entry 712 may be shared and may be used to translate virtual addresses for processors 710 and 740. Similarly a sharing indication corresponding to virtual address translation entry 719 indicates a shared status of S and a set of logical processes of 1111, indicating that entry 719 may be shared and used to translate virtual addresses for all four processors 710-780.

A sharing indication corresponding to virtual address translation entry 716 indicates a invalid status of I and a set of logical processes of 0000 meaning that entry 716 may not be used to translate virtual addresses for any processor 710-780. It will be appreciated that the invalid status may be explicitly represented or implicitly represented by the corresponding set of logical processes. It will also be appreciated that one skilled in the art may produce other encodings to explicitly or implicitly represent sharing indications for TLB entries.

The specification further states with regard to Fig. 7b (p. 20, par. 49-50, emphasis supplied):

In Figure 7b, for example, a sharing indication corresponding to virtual address translation entry 711 may implicitly indicate a private status of P and an explicit set of logical processes of 01 meaning that entry 711 may be used to translate virtual addresses for processor 710. It will be appreciated that such an implicit status representation may permit any implicit private status to be changed to an implicit shared status if another processor is found that may make use of the corresponding virtual address translation entry.

For example, if a processor initiates a TLB request to look up a virtual address translation and the sharing indication corresponding to the retrieved TLB entry indicates a set of logical processes that does not include one associated with the processor initiating the TLB request, then the physical address data and other TLB data may be recovered from page tables in main memory. Control logic 704 may include a mechanism for recovering such data, or may invoke a mechanism such as a page walker to access page tables in memory and compute physical addresses. If the newly constructed virtual address translation matches the retrieved TLB entry, the requesting process may be added to the set of logical processes sharing the retrieved TLB entry. Otherwise the newly constructed virtual address translation may be installed in a new TLB entry for the requesting processor.

Therefore, Applicant respectfully submits that the present application discloses how to make the identification of sharability of a translation transparent to the operating system such that one skilled in the art may practice the entire scope of the subject matter claimed without undo experimentation.

The Final Office Action states that the specification sections cited above do not mention that the translation is identified as sharable transparent to the OS. Applicant respectfully submits that the present specification further states (p. 16, par. 42, emphasis supplied):

It will be appreciated that control unit 604 provides for efficient sharing of TLB 602 entries among logical processes without requiring additional support from, or modifications to, any particular operating system that may be selected for use in conjunction with a multiprocessor or multithreading processor employing the apparatus of Figure 6 to provide sharing of virtual address translations in an address translation stage 502. One such multiprocessor or multithreading processor may, for example, execute a 32-bit Intel Architecture (IA-32) instruction set which comprises IA-32 instructions of the Pentium® processor family. Another such multiprocessor or multithreading processor may, for example, execute a 64-bit Intel Architecture (IA-64) instruction set which comprises IA-64 instructions of the Itanium™ processor family or may also execute a combination of both IA-32 and IA-64 instructions. Since such multiprocessors or multithreading processors may be used in various computer systems running any one of a number of

operating systems, an apparatus employed by such multiprocessors or multithreading processors to provide sharing of TLB entries should accordingly be operating-system transparent, providing sharing of TLB entries among logical processes without requiring that the operating system actively manage the sharing of all TLB entries. It will also be appreciated that if a multiprocessor or multithreading processor has a mechanism to provide sharing of TLB entries in such a way that is operating-system transparent or operating-system independent, that it does not prohibit that multiprocessor or multithreading processor from also providing for additional operating-system support for managing some sharing of TLB entries.

Therefore, Applicant respectfully requests the Examiner withdraw his rejections under 35 U.S.C. 112, first and second paragraphs.

35 U.S.C. § 102(e) Rejections

The Final Office Action rejects Claims 1-8, 13-30 and 36-46 under 35 U.S.C. 102(e) as allegedly being anticipated by U.S. Patent 6,598,050 B1 (Bourekas).

Applicant respectfully disagrees with the Examiner's anticipation assertions.

Applicant argued in the appeal brief filed May 11, 2004 that the invention of Bourekas relates to a virtual addressing scheme within a microprocessor based system (col. 1, lines 8-9). In this virtual addressing scheme, the virtual addresses have a group membership field (col. 2, lines 27-28). The group membership field is used to permit sharing of data and/or programs among a subset of tasks in a multi-tasking system (col. 2, lines 21-23). With the use of the group membership field, the operating system can support three levels of access in a virtual to physical address translation. The operating system permits a global translation, an individual translation and a group translation," (col. 5, line 65 through col. 6, line 2, emphasis added).

In response to the above argument, the Examiner argued that neither this nor anything else in the reference requires the operating system to be modified as required by the apparent definition of "transparent" in the claims and that Bourekas does not disclose how the operating system was to be modified.

Applicant respectfully submits that when the operating system is not modified to support virtual addresses that have the group membership fields of Bourekas, there is no alternative method to permit shared translations, either expressly or inherently described. Whether Bourekas is enabling is not relevant to the issue at hand. The MPEP § 2131 states that:

> "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal

Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

Claims 1 sets forth operating-system transparent methods that comprise accessing and transparently identifying if the virtual address translation is sharable. Claims 13 and 20 both set forth apparatus to provide operating-system transparent sharing of virtual address translations comprising control logic to transparently produce or provide a sharing indication if a virtual address translation may be shared.

Applicant respectfully submits that in the cited reference, each and every element as set forth in the independent claims 1, 13 and 20 is not found, either expressly or inherently described.

In the multithreading processor of claim 36, a control logic comprises circuitry to identify a sharability of a first TLB entry and to provide a first sharing indication to indicate if the first entry may be shared by a second process. Claim 36 further sets forth a sharing indication field in the first TLB entry to store the first sharing indication provided by the control logic.

Bourekas relates to a virtual addressing scheme wherein a group of tasks may be marked for access to a given translation (col. 1, lines 8-9; col. 3, lines 1-3). In Bourekas, both the global bit and the group membership field stored in the TLB entry come from the virtual address translation provided by the operating system (col. 5, line 65 through col. 6, lines 2 and lines 51-52)--not from processor control logic as set forth in claim 36. Matching circuitry of Bourekas receives static group membership fields from the virtual address and from the TLB and simply matches them to determine if they were marked as belonging to the same group (col. 7, line 60 through col. 8, line 2).

The MPEP § 2131 states that:

"The identical invention must be shown in as complete detail as is contained in the ... claim." Richardson v. Suzuki Motor Co., 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

Applicant respectfully submits that Bourekas does not expressly or inherently describe, in as complete detail as is set forth by claim 36: a multithreading processor with control logic to identify sharability of a TLB entry, to provide a sharing indication to indicate if the first entry may be shared by another process and to store the sharing indication provided by the control logic in a field in the TLB entry.

Accordingly in light of the argument presented above, Applicant submits that independent claims 1, 13, 20 and 36 are not anticipated by Bourekas.

The Office Action also rejects Claims 1-8, 13-30 and 36-46 under 35 U.S.C. 102(e) as allegedly being anticipated by U.S. Patent 6,564,311 B2 (Kakeda).

Applicant respectfully disagrees with the Examiner's anticipation assertions.

Kakeda relates to yet another virtual addressing scheme wherein a group of tasks may be marked for access to a given translation. With regard to Figures 4 and 5, Kakeda contrasts the conventional address translation using one global bit (Fig. 5) as compared with the two-global-bit embodiment of his invention (Fig. 4, col. 9, line 24 through col. 10 line 25). He discloses that his global bits are the "comparison information" (col. 2, lines 53-62) and describes how process identifiers (col. 8, lines 13-43) and global bits must be set (col. 8, line 48 through col. 9, line 13). The apparatus of Kakeda is specific to the addressing scheme described and therefore the establishment of page tables (col. 12, line 49, and Fig. 4) to take advantage of Kakeda is not operating-system transparent.

Accordingly in light of the argument presented above, Applicant respectfully submits that in the cited reference, each and every element as set forth in the independent claims 1, 13 and 20 is not found, either expressly or inherently described.

As stated above with regard to claim 36, a control logic comprises circuitry to identify a sharability, to provide a first sharing indication to indicate if the first entry may be shared by a second process and a sharing indication field in the first TLB entry stores the first sharing indication provided by the control logic.

With regard to the comparators 13 and the AND gates 105, Kakeda does not disclose providing a first sharing indication to indicate if the first entry may be shared by a second process as argued by the Examiner. Further, Kakeda does not disclose that what comparators 13 and AND gates 105 do provide is stored in process identifier storage area 102 or that what is stored in process identifier storage area 102 is provided by comparators 13 and AND gates 105 as argued by the Examiner (col. 5, line 66 through col. 6, line 34, and Figs. 1-2).

Accordingly in light of the argument presented above, Applicant submits that independent claims 1, 13, 20 and 36 are not anticipated by Kakeda.

CONCLUSION

Applicant respectfully submits the present claims for allowance.

Authorization is hereby given to charge our Deposit Account No. 02-2666 for any charges that may be due.

If the Examiner believes a telephone conference would expedite or assist in the allowance of the present application, the Examiner is invited to call Lawrence Mennemeier at (408) 765-2194.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Date: 12-19-05

Lawrence M. Mennemeier
Reg. No. 51,003

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8300